

Absolute Beginner's Python Programming

Solutions To Lab Exercises

Contents

Lab Exercises Chapter 2 Solutions.....	2
Lab Exercises Chapter 3 Solutions.....	4
Lab Exercises Chapter 4 Solutions.....	7
Lab Exercises Chapter 5 Solutions.....	8
Lab Exercises Chapter 6 Solutions.....	9
Lab Exercises Chapter 7 Solutions.....	11
Lab Exercises Chapter 9 Solutions.....	12
Lab Exercises Chapter 10 Solutions	15

Lab Exercises Chapter 2 Solutions

What is the output produced by the following code fragment?

```
num1 = 2
num2 = 3
print (num1 + num2)
```

5

What is the output produced by the following code fragment?

```
num1 = 2
num2 = 3
print ("num 1 + num 2 = ", num1 + num2)
```

num 1 + num 2 = 5

Find the errors in the following program

```
Num1 = 2
num2 := 3
Sum = num1 + num2;
printf(sum)
```

:= not an operator
Num 1 is declared above not num1
printf should be print and sum should be Sum

What is an identifier?

The name given to identify a variable, function, class, module or other object.

Which of the identifiers below is valid and which are invalid? Why?

```
Num1
time-of-day
tax_rate
x5
int
7th_Rec
yield
```

cant have a -
is a reserved word
can't start with a number
reserved word

How do you write comments in your code? Explain with an example.

Use # or ""

Why should you include comments?

For documentation purposes and to explain sections of code.

What is a reserved word?

Python words with a predefined meaning.

What is a low level and high level language?

A high-level language is one that is user-oriented and consists of 'english like' statements designed to make it easier for a programmer write code. These statements are then compiled or interpreted. A low-level language is machine-oriented.

What is object-oriented programming?

A computer programming model that is based around objects, rather than functions and logic.

Lab Exercises Chapter 3 Solutions

Write a program that accepts a length in inches and prints the length in centimeters (1 inch = 2.54cm).

```
inch = int(input('Enter the value in inches: '))
cm = 2.54 * inch
print('{} inches = {}cm'.format(inch, cm))
```

Note use float() for real numbers instead of int() when casting the data type in line 1, and add {:.2f} in string place holder for two decimal places

Write a program that accepts your forename, surname and year of birth and adds them to an array.

You can insert into the array at specific index

```
student = ['one', 'two', 'three', 'four']
student[0] = 'first name'
student[1] = 'second name'
student[2] = '1988'
print(student)
```

Or append to the end of the array

```
student = ['one', 'two', 'three', 'four']
student.append('first name')
print(student)
```

Write a program that adds some employee data to a dictionary. Use an employee number as the key.

```
employee = {
    '100': 'Anne',
    '101': 'John',
    '102': 'Emma'
}
employee.update( {'103': 'John'} )
print (employee)
```

Write a program that converts temperatures from Celsius to Fahrenheit.

$$F = C \times 9/5 + 32$$

```
fah = int(input('Enter the temperature in celcius: '))
cel = fah * 9/5 + 32
print('{} degrees C = {} Fahrenheit'.format(fah, cel))
```

Write a program that calculates the volume of a sphere

$$V = \frac{4}{3} \pi r^3$$

Import the math library to use Pi

```
import math
r = int(input('Enter the radius: '))
v = 4/3 * math.pi * r ** 3
print('Volume is {}'.format(v))
```

Write a program to calculate and display an employee's gross and net pay. In this scenario, tax is deducted from the gross pay at a rate of 20% to give the net pay.

```
gross = float(input('Enter the gross pay: '))
net = gross - gross * 20/100
print('Gross Pay {:.2f} Net Pay {:.2f}'.format(gross, net))
```

Write a program that stores a shopping list of 10 items. Print the whole list to the screen, then print items 2 and 8.

```
shoppingList = ['bread', 'milk', 'coffee', 'sugar', 'cereal', 'veg', 'beans', 'rice',
                'pasta', 'onions']
print(shoppingList[2])
print(shoppingList[8])
```

What does it print? Remember the list starts from 0 not 1.

```
coffee
pasta
```

Extend the previous program, to insert an item into the list.

```
shoppingList [2] = 'tea'
or append to end
shoppingList.append('ham')
```

What is a Boolean operator? Write a program to demonstrate.

Logic operator such as AND, NOT, OR. Commonly used to combine conditional statements (eg loops & if-else statements)

```
print (x < 10 and x < 20)
```

What is a comparison operator? Write a program to demonstrate.

Compares on value with another. More commonly used in loops and if else statements

```
x = 10
y = 14
print (x > y) #prints false as x is not greater than y
```

What is data type casting? Why do we need it? Write a program to demonstrate.

Convert a variable data type from one to another. Python converts data type into another data type automatically (implicit) depending on what value is assigned to the variable: string, int, etc. If you need to change the type using eg int(). This is explicit.

```
a = 2.2          #python casts as float
int(a)          #change to integer
```

Lab Exercises Chapter 4 Solutions

Write a program to print the numbers 1 - 10 to the screen.

```
for counter in range(1, 11):  
    print("We're on %d" % (counter))
```

Write a program to print a list of names to the screen.

```
lists = ['name 1', 'name 2', 'name 3', 'name 4', 'name 5', 'name 6']  
for counter in lists:  
    print (counter)
```

Write a program to calculate and print the squares of the numbers from 1 to 10. Use \t if you want to display in a table.

```
for counter in range(1, 11):  
    print("Square of %d is %d" % (counter, counter*counter))
```

Write a program that accepts a number from the user until a negative number is entered.

```
userinput = 1  
while userinput > 0:  
    print ("Enter a number: ")  
    userinput = int(input())  
print ("ok")
```

Write a program that accepts an integer and prints the specified range it belongs to.

Range 1: 0 to 10

Range 2: 11 to 20

Range 3: 21 to 30

Range 4: 31 to 40

```
num = int(input("Enter number: "))  
if num > 40 :  
    print('Out of Range')  
elif num >= 30 :  
    print('Range 4')  
elif num >= 20:  
    print('Range 3')  
elif num >= 10:  
    print('Range 2')  
elif num >= 0:  
    print('Range 1')  
else:  
    print('Out of range')
```


Lab Exercises Chapter 5 Solutions

Write a program that gets a string from the user then writes it to a file along with the user's name.

```
name = input("Enter your name: ")
file = open('names.txt', 'w')
file.write(name)
file.close()
```

Modify the program from exercise 1 so that it appends the data to the file rather than overwriting.

```
name = input("Enter your name: ")
file = open('names.txt', 'a')
file.write(name)
file.write("\n")
file.close()
```

Write a program to write a list of names to a file.

```
file = open('file1.txt', 'w')
names = ['name 1', 'name 2', 'name 3', 'name 4']
for index in names:
    file.write("%s\n" % index)
file.close()
```

Write a program to read a file line by line and store it in a list.

```
file = open('file1.txt', 'r')
count = 0
for line in file:
    count += 1
    print("{}".format(line.strip()))
file.close()
```

What is the difference between a text file and a binary file?

- Binary files typically contain a sequence of bytes and are much smaller and faster than text files.
- Text files store data in ASCII format and are plain text meaning they have no formatting and are human readable. For example .txt, .py, .HTML, etc.
- Binary files store data in binary format and are usually Images, Word and Excel documents, Videos, Applications (exe), etc.

Lab Exercises Chapter 6 Solutions

Write a program that accepts a number from the user and uses a function to square the number then return the result.

Print the result to the screen.

```
def sq(a):  
    return a * a  
num = int(input("Enter a number: "))  
print (sq(num))
```

Write a function that returns the largest of two numbers. Test the function and print the results to the screen.

```
def largest(a, b):  
    if a > b:  
        return a  
    return b  
num1 = int(input("Enter number one: "))  
num2 = int(input("Enter number two: "))  
print (largest(num1, num2))
```

What is the difference between a local and a global variable?

Global is accessible from anywhere, local is only accessible within the function.

Write a program that prints first 10 positive numbers using a recursive function.

```
def display(num):  
    if(num) :  
        display(num-1)  
    else :  
        return  
    print("{}".format(num))  
  
limit = int(input("Enter a number: "))  
print("\nNatural Numbers from 1 to {} are:".format(limit))  
display(limit)
```

What's the difference between a parameter and an argument?

A parameter is a variable in the function definition. An argument is a value passed during a function call.

What is a built in function?

A function that has been defined within the python programming language

What is a user defined function?

A function that is defined by the programmer

What makes a function recursive?

The function calls itself.

Advantages and Disadvantages of Recursion

Recursive programs allow programmers to write efficient functions using a minimal amount of code and is useful technique that can reduce the length of code and make it easier to read and write. However, if performance is vital, it is better to use iteration, as recursion can be a lot slower.

Lab Exercises Chapter 7 Solutions

Write a program that accepts a number from the user and uses a function to square the number then return the result.

```
def sq(num1):  
    return num1 * num1
```

Save this file as a module

```
myfunctions.py
```

Import the module you just created into a new program.

```
import myfunctions
```

Call the function in the module

```
userInput = int(input("Enter a number: "))  
result = myfunctions.sq(userInput)  
print(result)
```

Lab Exercises Chapter 9 Solutions

Declare a new class called Vehicle without any attributes and methods

```
class Vehicle :
```

Add some attributes to the Vehicle class such as Name, Speed, Mileage

```
class Vehicle :
    def __init__(self, name, speed, mileage):
        self.name = name
        self.speed = speed
        self.mileage = mileage
```

Add a method to the Vehicle class to return the vehicle name

```
class Vehicle :
    def __init__(self, name, speed, mileage):
        self.name = name
        self.speed = speed
        self.mileage = mileage
    def getName(self):
        return self.name
```

Create a child class called Taxi

```
class Taxi (Vehicle):
    def __init__(self, name, speed, mileage):
        super().__init__(name, speed, mileage)
```

Add a method to the Taxi class to collect the fare. For example, fare is calculated by mileage * 0.20

```
class Taxi (Vehicle):
    def __init__(self, name, speed, mileage):
        super().__init__(name, speed, mileage)
    def getFare(fare):
        self.fare = mileage * 0.20
```

Let's test it...

Get the data from the user and assign to variables rn and mil

```
rn = input("Enter route name: ")
mil = int(input("Enter Mileage: "))
```

Create object then pass data we captured from the user (name, speed, mileage)

```
route = Taxi(rn, 0, mil)
```

Call object method to calculate fare, pass the mileage captured from user to the getFare() method

```
route.getFare(mil)
```

Print results

Use the getName() method to get return the name. We can just reference the attribute using the dot notation (route.mileage, and route.fare)

```
print("\n\nRoute name is %s" % (route.getName()))
print("Mileage is %d" % (route.mileage))
print("The fare is $%f" % (route.fare))
```

When we run the program, we get this output

```
Enter route name: test route
Enter Mileage: 12
```

```
Route name is test route
Mileage is 12
The fare is $2.400000
```

(12 x 0.2 = 2.4)

What is a class? Show an example

A template for creating objects

```
class Person :
    ...
    ...
```

What is an object? Show an example.

An instance of a class

```
John = Person (...)
```

What is a constructor? Show an example.

A method that is called when an object is created.

```
def __init__(self, name, dob, email):
```

What is a method? Show an example.

What is `__init__()` used for? Show an example.

To initialise the object

```
def __init__(self, name, dob, email):
```

What is `super()` used for? Show an example.

The `super()` function calls the `__init__()` method of parent class

```
super().__init__(name, dob, email)
```

Lab Exercises Chapter 10 Solutions

Create a new program and import the turtle graphics module.

```
import turtle
```

Experiment with drawing different shapes using some of the turtle graphics methods. Use the turtle commands to draw some shapes.

```
import turtle
turtle.pensize(6)
turtle.penup()
turtle.pendown()
for i in range(5):
    turtle.forward(200)
    turtle.right(144)
turtle.done()
```